Projekt

Teoria i metody optymalizacji

Harmony Search

Autor:
Artur Gasiński, 218685
Bartosz Lenartowicz, 218518

Termin: wt 9:15-11:00

Prowadzący:
Dr inż. Ewa Szlachcic

20 maja 2018

Spis treści

1 Wstęp

Większość metod optymalizacji procesów produkcyjnych czy logistycznych sprowadza się do przedstawienia zależności pomiędzy procesami, którymi firma jest w stanie manipulować, za pomocą zmiennych. Następnie tworzone jest równanie matematyczne opisujące system firmy. Takie równanie nosi nazwę funkcji celu. By właściwie zoptymalizować proces należy znaleźć, w zależności co poddane jest optymalizacji, minimum bądź maksimum globalne takiej funkcji. Obliczanie minimalnej bądź maksymalnej wartości funkcji, zależnej od wielu zmiennych w pożądanym czasie, nie jest zadaniem łatwym. Istnieją specjalne algorytmy do rozwiązywania zadań tego typu. Niniejsza praca została napisana by przedstawić jeden z takich algorytmów – Harmony Search. Wpierw w rozdziale 2 został przedstawiony opis tego algorytmu. Rozdział 3 przedstawia implementacje Harmony Search w programie komputerowym. (...)

2 Opis działania algorytmu

Algorytm Harmony Search został przedstawiony w 2001 przez Zong Woo Geem, Joong Hoon Kim oraz G.V. Loganathan w pracy Ą New Heuristic Optimization Algorithm: Harmony Search" [4]. Inspiracją dla algorytmu było szukanie przez muzyków jazzowych podczas improwizacji najlepszych harmonii dźwięków. Umożliwia on znajdowanie minimów lokalnych funkcji wielu zmiennych. Główna zasada polega wyszukiwaniu rozwiązań na podstawie wartości wcześniej obliczonych oraz modyfikacji nowych zmiennych z określonym z góry prawdopodobieństwem. Szczegółowe przedstawianie algorytmu mija się z celem ponieważ przebieg algorytmu został przetłumaczony na język polski np. w pracy [2]. W następnym rozdziale 3 zostanie przedstawiona szczegółowo implementacja algorytmu.

3 Implementacja

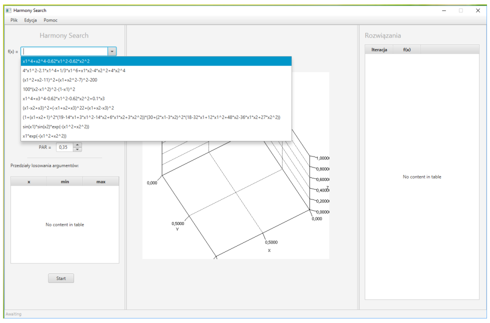
Program do obliczania minimum funkcji na podstawie algorytmu Harmony Search został napisany w języku Java. Technologią dostarczającą GUI dla użytkownika była *JavaFx*. Program korzystał również z zewnętrznej biblioteki do obliczania wartości funkcji w punkcie [3]. Wykres prezentujący najlepsze rozwiązanie zaznaczone na wykroju z przestrzeni rozwiązań funkcji został stworzony za pomocą biblioteki *jzy3D* [1].

3.1 GUI

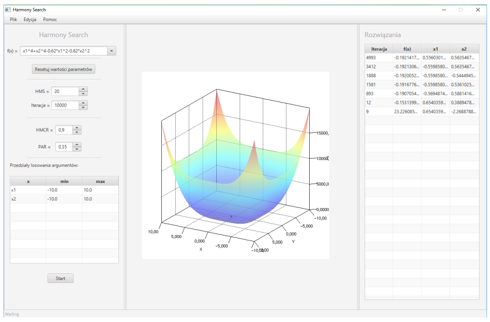
By ułatwić wprowadzanie funkcji do programu oraz bezproblemowo odczytywać wyniki obliczone przez program zostało stworzone GUI programu. Na rysunkach 1, 2 i 3 przedstawiono etapy działania programu. Poszczególne elementy GUI podczas działania programu nie zmieniają swojego położenia. Jedynie pustych w miejscach zostają wyświetlone ich parametry, dlatego by omówić elementy posłużono się rysunkiem 3, gdzie widnieje obraz programu po obliczeniach. Główne okno programu zostało podzielone pionowo na trzy części. Pierwsza część od lewej przeznaczona jest do wpisania funkcji i jej parametrów, środkowa prezentuje wykres, a część prawa w tabelce wyniki obliczeń programu. Wszystkie części zostaną szczegółowo opisane poniżej.

3.1.1 Parametry

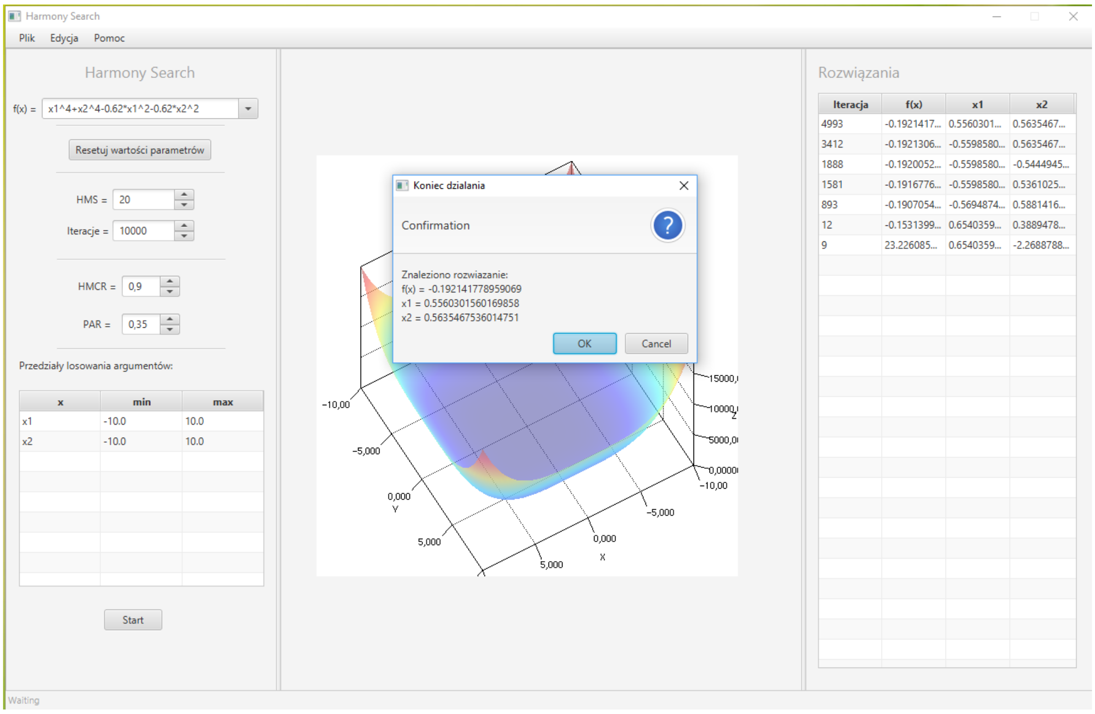
W części parametrów można dostrzec, że zaraz pod paskiem edycji znajduje miejsce do wpisania funkcji. Jest to ta sama funkcja której minimum należy wyznaczyć. Okno umożliwia również wybór takiej funkcji z wcześniej zadeklarowanych w programie. Dokonuje się tego klikając w rozwijane menu. Zostało to przedstawione na rysunku 1. Po wybraniu funkcji automatycznie zostaną domyślnie uzupełnione wszystkie parametry obliczeń. Oczywiście możliwa jest dostosowanie algorytmu do własnych potrzeb. Poniżej okna funkcji znajduje się przycisk „`Resetuj wartości parametrów`” umożliwiający, jak mówi nazwa, reset do parametrów domyślnych. Poniżej przycisku zostały umieszczone po sobie cztery główne parametry algorytmu *Harmony Search*.
Pierwszy z parametrów `HMS` (*ang. harmony memory size*) reguluje ilość zapamiętywanych rozwiązań podczas wyszukiwania. Domyślnie rozmiar jest zadeklarowany na 20. Poniżej znajduje się okno o nazwie



Rysunek 1: Wpisywanie funkcji



Rysunek 2: Wykres z zaznaczonym rozwiązaniem



Rysunek 3: Obliczone rozwiązanie wraz z komunikatem

`iteracje`. Domyślnie jest zadeklarowana maksymalna wartość iteracji równa 10000. W każdej iteracji programu algorytm wyszukuje nowe rozwiązanie i sprawdza je z tymi zapisanymi w `HMS`. Czym większa ilość iteracji tym algorytm powinien obliczać dokładniejsze rozwiązanie, kosztem wydłużonego czasu obliczeń. Następnie pod polem `iteracje` znajduje się pole `HMCR` (*ang. harmony memory consideration ratio*). Tłumaczenie określa go jako współczynnik wyboru tonu z pamięci. Jest to współczynnik prawdopodobieństwa wybierany z zakresu $(0, 1)$. Czym `HMCR` jest większe tym wyszukiwane wartości x następnych rozwiązań będą zbliżone do tych istniejących już w tablicy `HMS`. Poniżej znajduje się drugi parametr prawdopodobieństwa dla funkcji o nazwie `PAR` *(ang. pitch adjustment ratio)* tłumaczony jako współczynnik dostosowywania tonu. Wybierany jest również z zakresu $(0, 1)$. Na samym dole kolumny znajduje się tabela `Przedziały losowania argumentów`. W tabeli deklarowany jest przedziały wszystkich wartości x dla których ma być poszukiwane rozwiązanie. Domyślnie parametry dla wszystkich x brane są z przedziału $(-10, 10)$. Algorytm z dużą dokładnością znajdzie najmniejsze minimum lokalne dla funkcji z podanego przedziału. Po wybraniu wszystkich parametrów można przystąpić do uruchomienia algorytmu przyciskiem `Start` znajdującym się na na dole kolumny.

3.1.2 Wykres

Środkowy obszar okna głównego przeznaczony jest do wyświetlania wykresu funkcji. Jak wspomniane zostało na początku rozdziału 3, do rysowania wykresów została użyta zewnętrzna biblioteka *jzy3D*. Umożliwia ona tworzenie wykresów dwu i trzy wymiarowych. Wykresy można obracać w dowolny sposób. Oś X i Y wykresu wyskalowana jest do najmniejszego i największego parametru wszystkich wartość x. Dodatkowo wartości zostały pokolorowane w tak, że mniejsze wartości funkcji mają kolory zimniejsze, a wyższe kolory cieplejsze, analogicznie do tego jak przedstawia się wysokość n.p.m. w kartografii. Dodatkowo czarną kropką zaznaczona została najmniejsza obliczona wartość funkcji.

3.1.3 Tabela Rozwiązań

Po prawej stronie okna głównego można dostrzec najlepsze rozwiązania wyszukane przez program. Gdy znajdowane jest rozwiązanie lepsze od najlepszego rozwiązania zapamiętanego w `HMS` dodawane jest ono jako pierwszy wiersz w tabeli. Poprzednie rozwiązania przesuwane są o jedno miejsce niżej. Pozwala to śledzić jaka wartości jest aktualnie największa oraz od razu ukazuje się obok numer iteracji tego rozwiązania. Gdy algorytm długo oblicza najlepszy wynik okno pozwala śledzić postęp obliczeń. Dodatkowo po wykonaniu wszystkich iteracji zostaje wyświetlony komunikat z najlepszym rozwiązaniem (pierwszym od góry), co przedstawia rysunek 3.

4 Przykładowe rozwiązania

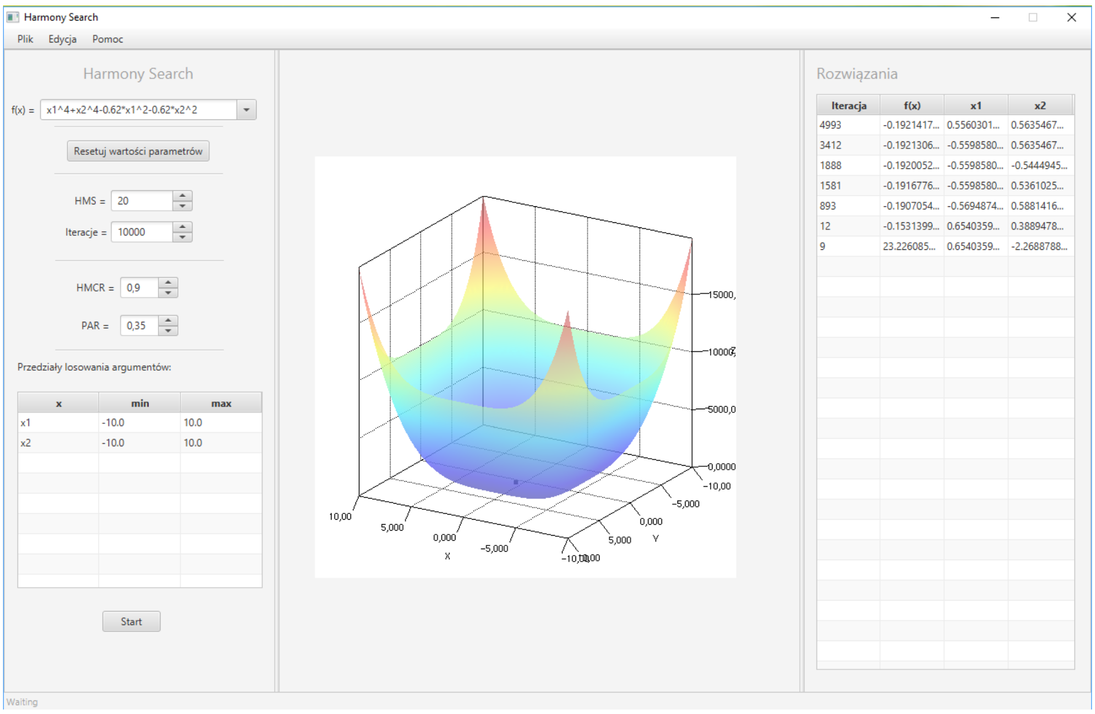
W tej części zostały przedstawione trzy przykłady funkcji dwóch zmiennych x_1 i x_2 wraz z obliczonymi rozwiązaniami.

4.1 Funkcja czterech minimum

Pierwsza zostanie przedstawiona funkcja posiadająca cztery minima lokalne. Posiada ona następujący wzór:

$$f(x) = x_1^4 + x_2^4 - 0,62x_1^2 - 0,62x_2^2$$

Na przykładzie tej funkcji zostało również omówione GUI programu w rozdziale 3.1. Rysunek 4 prezentuje wykres funkcji wraz z najlepszą wartością. Dokładną wartość można odczytać po prawej stronie i jest to 4993 iteracja równa $f(0.5560, 0.5635) = -0.1914$. Poniżej zaprezentowany jest wykres funkcji.



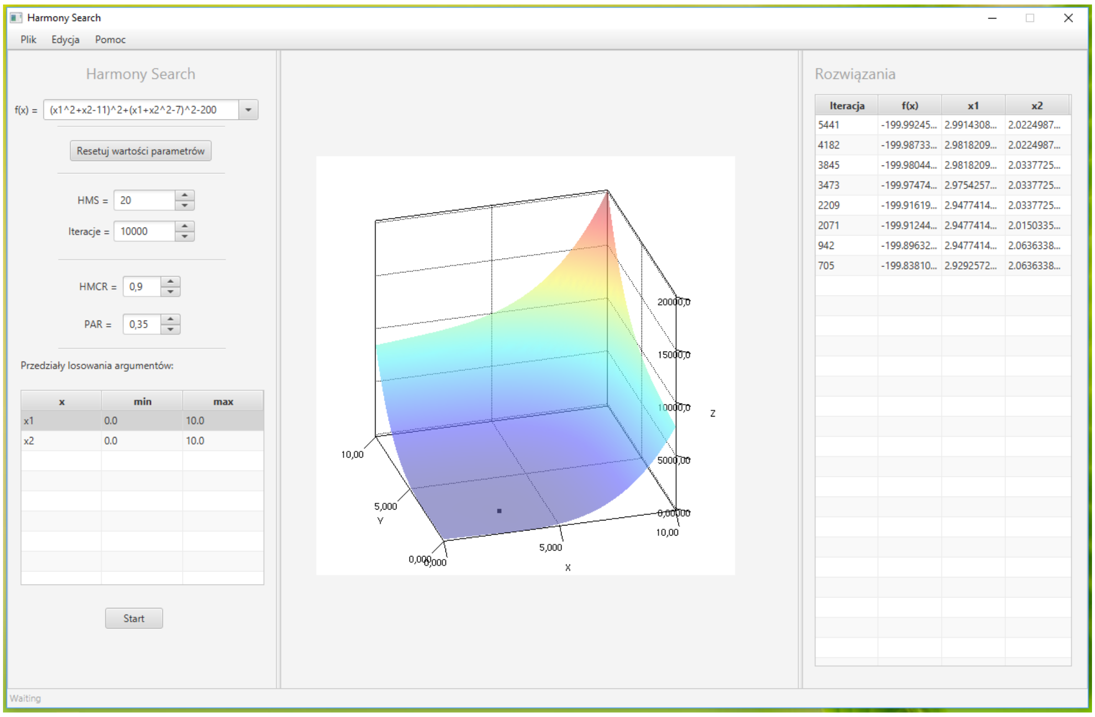
Rysunek 4: Wykres funkcji czterech minimum

4.2 Funkcja Himmelblau

Funkcja Himmelbalu również jak 4.1 posiada cztery minima lokalne zlokalizowane po jednym w każdej ćwiartce na przedziale $x_{1,2} \in (-5, 5)$. Funkcja posiada wzór:

$$f)x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 - 200$$

. Dla pokazania, że funkcja wyszukuje rozwiązania z innego przedziału jak domyślny ograniczono się do przedziału z pierwszej ćwiartki $x_1, x_2 \in (0, 10)$. Na wykresie 5 zaprezentowany został wykres funkcji wraz z naniesionym najlepszym rozwiązaniem.



Rysunek 5: Wykres funkcji Himmelblau

4.3 Funkcja sinus-cosinus z eksponentom

Ostatnia przedstawiana funkcja ukazuje przypadek połączenia funkcji sinusa-cosinusa z eksponent. Przypadek jest o tyle ciekawy, że większość wartości funkcji jest dla x_1, x_2 jest do siebie zbliżona. Blisko jej wartość ulega dużemu odchyłowi. Wzór funkcji przedstawiony jest jako:

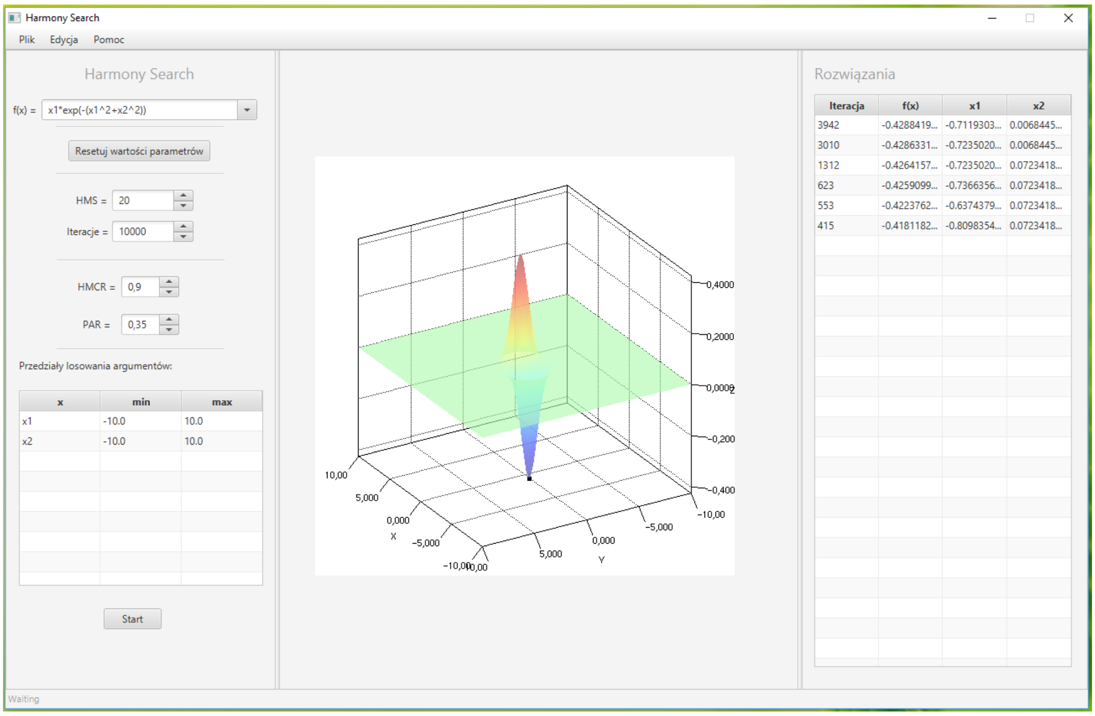
$$x_1 e^{-(x_1^2 + x_2^2)}$$

. Rysunek 6 prezentuje działanie programu z obliczoną wartością minimalną. Jak możemy dostrzec program dobrze wyliczył jej wartość co potwierdza jego odporność na działanie funkcji, dla której większość argumentów jest stała.

5 Problemy podczas pracy

Jednym z problemów z jakim musiał poradzić sobie program było prezentowanie danych. Algorytm jest zdolny do obliczania funkcji wielu zmiennych dla $x_n, n \in N$. Prezentowanie zmiennych w tabelach jest rzeczą dającą się w prosty sposób rozwinąć. Jednak rysowanie wykresu dla $n \leq 2$ ilości zmiennych nie jest możliwy. Najładniejsze wykresy powstają dla $n = 2$ dlatego funkcje o dwóch zmiennych zostały przedstawione w rozdziale 4.

6 Podsumowanie



Rysunek 6: Wykres funkcji sinusa-cosinusa z eksponentom

Literatura

[1] Jzy3d. http://www.jzy3d.org/, 2016.

[2] Kowal Andrzej. Algorytm „harmony search”.

[3] Mariusz Gromada. Mathparser. https://mvnrepository.com/artifact/org.mariuszgromada.math/MathParser.org-mXparser, 2017.

[4] G.V. Loganathan Kang Seok Lee, Zong Woo Geem. A new heuristic optimization algorithm: Harmony search, 2005.